Project spezification to the course „Big Data Analytics“

Implementation of a Data Science project across all 4 layers (Data Source Layer, Data Storage Layer, Processing Layer, Data Output Layer)

Topic: the exact topic can be chosen by yourself

Procedure & general conditions

1. Teamwork

Each project is worked on **by 2 students** (teamwork)

You will be evaluated as a group.

2. Big Picture

In this project, you will learn how to independently carry out Data Science projects with a focus on methods as applied to Big Data. Infrastructural issues in particular will be considered. Then you continue with the exploration and collection of suitable data from public sources or you use your own data. Only those who know their data can evaluate them in a meaningful way. Therefore, an intensive examination of the data and its description and documentation is indispensable. The data must be checked for errors, cleaned and combined in a suitable form. Initial visualizations provide information about the structure and distribution of the data so that further steps can be derived from them. As a result, the cleaned data must be stored in a suitable form. With the help of Data Science methods, an "added value is generated" from the given data. The result must be presented in a clearly understandable form (visualization, story telling).

Note: in most cases, the project will not actually fall into the Big Data category due to the relatively small amount of data, but similar procedures will still be used.

3. Project Structure

The project is divided into a total of 4 parts (based on the 4 levels of Big Data). All 4 levels should be clearly visible, even if (depending on the topic), the individual levels can be prioritized very differently:

a) **Data Source Layer**: Collection of all data of interest for later analyses → Identification of data sources, methods of data extraction, Open Data, ready-made datasets, WebScraping, …
b) **Data Storage Layer**: Store your data in a suitable form (SQL, NoSQL, FlatFiles, Streaming, …)
c) **Analysis Layer**: Process your data, create additional value (data analytics, time series analytics, text analytics, machine learning, …)
d) **Data Output Layer / Visualization**: present the newly added value (visualization, story telling)

Parts a) to d) must be documented as well as possible. **Jupyter notebooks** are used as the form of **documentation**. Ultimately, a mixture of code, description, documentation, processes and results is created.

4. Delivering results

For a DataScience project it is inevitable to deal intensively with the topic. Both topic and project goal are not clearly defined, let your imagination run wild and make something "vivid" out of the data you have available. Present "**your story**" as vividly as possible. Good examples of this are given by Kaggle projects, as for example:

- Specification & data: https://www.kaggle.com/c/kaggle-survey-2019
- Example solutions: https://www.kaggle.com/parulpandey/geek-girls-rising-myth-or-reality/notebook

In order to implement the individual steps you have to …

- conduct supplementary research on the topic
- produce the technological bases and understand their functioning (through manual study complementary to the course)
- implement the task in a showcase
- present the present the results, methods and approaches to the whole group (approx. 30 minutes per topic) at the end of the course
- In addition, you create HOW-TOs (=documentation) in the form of Jupyter Notebooks

Milestones:

- class 4: Submit your topic (short talk in the unit)
  - topic (title)
  - members (team)
  - planned data sources
  - planned data storage
  - planned procedure
  - expected output
- class 6: intermediate delivery:
  - brief discussion during the attendance phase on the status of your project
- class 8: final delivery:
  - all documents in Moodle
  - presentation of the results (in a team, 30 min)

5. Infrastructure

The starting point of your work is to build the necessary infrastructure and tools:

Install and experiment with various Python libraries. Deploy the infrastructure on your notebook or other machine, either natively, but also gladly using Docker or a cloud system. The following points are recommended but optional (depending on the infrastructure you want to use)

- Install Docker (optional)
- Install a Python environment (Anaconda is recommended)
- Install additional Python libraries, like pySpark (including required Hadoop parts)
- Use Git as project repository (for backup and code sharing in the team)

A functional installation either locally or in a container or in the cloud is expected.

6. How to write Jupyter Notebooks?

The aim is to document your work briefly, clearly and comprehensibly. The primary focus here is on clarity and comprehensibility

Jupyter notebooks should not only be traceable, but also executable. This also means that the underlying data must also be delivered. If libraries beyond the standard scope of delivery of Python > 3.7 are needed, they have to be imported at the beginning of the file.

Note: It makes more sense to implement the exercise parts in smaller steps

A description of the Markdown syntax in Notebooks can be found e.g. in Jupyter Notebooks resp. Jupyter Lab directly.

7. Assessment

Depending on the selected topic, the 4 required levels can be prioritized differently. Nevertheless, parts of each level must be visible in some form. The points are then assigned according to the following key:

part	what to include (with examples)	points

Data Source	• use at least 2 different data sources
	• data identified, documented (what data do you have, how is it structured and organized)
	• some Webscraping or API part is obligatory
	• make data available
	• describe your data, which metadata does exist
	• examples:
	  ○ use ready datasets (e.g. Open Data Austria, Kaggle)
	  ○ use data from Web-APIs (e.g. OpenWeatherMap)
	• Checking data quality (completeness and validity of values, processing inconsistencies, recognizing correlations and dependencies, …)
	• first (statistical) analysis of raw data
	• …
		5

Data Storage	• use one or more databases or FlatFiles • the use of a NoSQL aspect is obligatory • communicate with DB (Import / Export / Python Scripts) • provide a suitable interface • Exploitation of specific properties of the database used • merge multiple data sources appropriately	5
Data Analytics	• Approach of analysis • Inventiveness and methods / procedures used • Further questions, usefulness of the answers found • Feature extraktion for Machine Learning	5
Data Presentation	• clean, coherent presentation of the results (the story is clearly recognizable) • appropriate visualization methods used • Layout, structure and story in Jupyter Notebook	5
Big Data criteria	• Similar to the analysis of the Connected Car video according to Big Data criteria. • Consider your project project according to the Big Data Vs (Volume, Velocity, Variety, Veracity, Value). Argue for each point the implications to your project idea. • Consider your project according to the 4 Levels of Data Handling in Data Science (Data Source, Data Storage, Data Analysis, Data output). Argue for each point the implications to your project idea.	5
Documentation	• in the form of a Jupyter Notebook • list all libraries you use (including their version) • list the sources of your data • describe everything that is necessary to know (tools, usernames & passwords – except those that are really private like API keys, …)	5
Components to be used	• the following components / tools have to be used in some way (if not, points are deducted) • NoSQL, WebScraping, Spark, MapReduce (either as pure MapReduce or with Spark), machine learning (at least something simple), Jupyter Notebooks	5
Quality in general	• overall impression of the project • how do the individual points interlock (do they give the impression of an overall project or are they rather independent partial solutions?) • everything that doesn't fit to above points	10
Präsentation	• presentation, talk, adherence to deadlines, … • everything that doesn't fit to above points	5
	sum:	50